■ Count Zeros & Ones

■ Problem Statement

You are given an array consisting only of **0s** and **1s**.
Your task is to count how many zeros and ones are present.

Example 1:
nums = [0, 1, 0, 1, 1, 0] → zeros: 3, ones: 3

Example 2:
nums = [1, 1, 1, 1, 1] → zeros: 0, ones: 5

Example 3:
nums = [0, 0, 0, 0, 0, 0] → zeros: 6, ones: 0

■ Core Logic

- Loop through the array.
- If value is 0 → increment totalZeros.
- If value is 1 → increment totalOnes.

Return both values using a pair.

■ Dry Run (Example: [0,1,0,1])

Index	Value	Zeros Count	Ones Count
0	0	1	0
1	1	1	1
2	0	2	1
3	1	2	2

■ Time & Space Complexity

Time Complexity: O(n)
Space Complexity: O(1)

■ Clean C++ Code

```
class Solution {
public:
  std::pair countZerosAndOnes(const std::vector &nums;) {
    int totalOnes = 0;
    int totalZeros = 0;
    int size = nums.size();

    for (int i = 0; i < size; i++) {
      if (nums[i] == 0) totalZeros++;
      if (nums[i] == 1) totalOnes++;
    }

    return {totalZeros, totalOnes};
  }
};
```

Key Notes

- Simple and efficient solution.
- Requires only one loop.
- Works for any binary array.